Database Schema: Final Review Challenge Copy

SCENARIO

You have been hired as a **junior database designer** for **TechEase Solutions**. Your team is building a **Learning Management System (LMS)** for schools. Your project lead has given you **sample raw data** and a **draft schema** that needs improvement.
You will work step-by-step to:

1. Identify key schema components from raw data.
2. Detect problems in the draft schema.
3. Create the final logical schema diagram.
4. Apply normalization or relationship refinement.

OBJECTIVES

By the end of this activity, you will be able to:
✅ Identify the **components** of a database schema from raw data.
✅ Detect and fix design flaws in a schema.
✅ Create a **finalized logical schema diagram** independently.
✅ Apply **table normalization or relationship analysis** to improve design.

TOPICS TO APPLY

- Entities (Tables)
- Attributes (Columns)
- Data Types
- Primary Keys
- Foreign Keys
- Relationships (1:1, 1:M, M:N)
- Normalization (1NF, 2NF, 3NF)

ACTIVITY FLOW (Individual Work)

Part 1 – Extract Schema Components from Raw Data (15 minutes)

You are given a **raw data table** from the LMS prototype:

Student_ID	StudentName	ContactNumber	SubjectCode	SubjectName	TeacherID	TeacherNam
S001	Anna Reyes	9171234567	MATH101	Basic Mathematics	T01	Mr. Cruz
S001	Anna Reyes	9171234567	ENG101	English Grammar	T02	Ms. Santos
S002	Ben Castillo	9179876543	MATH101	Basic Mathematics	T01	Mr. Cruz
S003	Carla Dominguez	9174561234	SCI101	General Science	T03	Mr. Villanueva

Tasks:

1. Identify the **entities (tables)** you think should exist.
2. List the **attributes** for each table.
3. Suggest the **primary key** for each table.
4. Identify at least **one relationship** and label it (1:1, 1:M, or M:N).

Answer Here:

Entities	Attributes (Columns)	
Students	Student_ID, StudentName, ContactNumber, SubjectCode	
Subjects	SubjectCode, SubjectName, TeacherID	
Teachers	TeacherID, TeacherName, SubjectCode	

4. The Table **Students** and Table **Subjects** have a relationship of **1:M**

Part 2 – Review and Fix the Draft Schema (20 minutes)

Your project lead gives you a **draft schema**:

Student: Student_ID, FullName, ContactNumber, Subjects
Subject: SubjectCode, SubjectName, TeacherID

Teacher: TeacherID, FullName, ContactNumber, Subjects

Tasks:

1. Compare this draft with your **Part 1** work.
2. List **three (3) problems** you see in the draft.
    - Problem 1:
    - Problem 2:
    - Problem 3:
3. Suggest a fix for each problem.
4. Rewrite the corrected schema based on your fixes:

Problem No.	Problem Description	Suggested Fix
1	There is no **FullName** attribute given	Specify it depending on the role (e.g StudentName & TeacherName).
2	There is no **Subjects** attribute given, and **Subjects** must be a table	Remove the "**Subjects**" and replace it with **SubjectCod**
3	There are two attribute of "**ContactNumber**" which will lead to confusion.	Specify it depending on the role.(e.g TeacherContactNumber, etc

Code block

```
Table: Students
Columns: Student_ID, StudentName, StudentContactNumber, SubjectCode

Table: Subjects
Columns: SubjectCode, SubjectName, TeacherID

Table: Teachers
Columns: TeacherID, TeacherName, TeacherContactNumber, SubjectCode
```

Part 3 – Final Logical Schema & Sample Data (30 minutes)

Instructions:

Using your corrected schema from Part 2, do the following for **each table** you designed:

1. Complete the table structure: list attributes with data types, primary key(s), and foreign key(s).
2. Create **3–4 example rows** of dummy/sample data showing how records would look in each table.
3. For relationships, briefly note which foreign key links to which primary key in another table.

Tables:

Table Name	Attributes (with data types)	Primary Key(s)	Foreign Key(s)	Relationships
Students	Student_ID(VARCHAR) StudentName(VARCHAR) StudentContactNumber(INTEGER) SubjectCode(VARCHAR)	Student_ID	SubjectCode	Subjec
Subjects	SubjectCode(VARCHAR) SubjectName(VARCHAR) TeacherID(VARCHAR)	SubjectCode	TeacherID	Studen
Teachers	TeacherID(VARCHAR) TeacherName(VARCHAR) SubjectCode(VARCHAR)	TeacherID	SubjectCode	Subjec

Sample Data:

Students Table:

Student_ID	StudentName	ContactNumber	SubjectCode
SSJ1	Naruto	916545647	CS 002
SSJ2	Sasuke	949887465	PE 900
SSJ3	Sakura	974897647	THESIS 100

Subjects Table:

SubjectCode	SubjectName	TeacherID
CS 002	Introduction to Hacking	SENSEI 001
PE 900	Taijutsu Training	SENSEI 002
THESIS 100	Reality or Dream	SENSEI 003

Teachers Table:

TeacherID	TeacherName	SubjectCode
SENSEI 001	Gojo Satoru	CS 002
SENSEI 001	Gojo Satoru	MATH 404
SENSEI 002	Hatake Kakashi	PE 900
SENSEI 002	Rayleigh	THESIS 100

Example (Student table):

Table Name	Attributes (with data types)	Primary Key(s)	Foreign Key(s)	Relationships
Student	Student_ID (VARCHAR), StudentName (VARCHAR), ContactNumber (VARCHAR)	Student_ID	—	Enrollment (1:M)

Sample Data:

Student_ID	StudentName	ContactNumber
S001	Anna Reyes	9171234567
S002	Ben Castillo	9179876543
S003	Carla Dominguez	9174561234

Part 4 – Self-Check & Improvement (15 minutes)

Review your diagram using this checklist:
✅ Each table has a **primary key**
✅ Foreign keys are used for relationships
✅ Attributes are **atomic** (no multiple values in one field)
✅ Table and attribute names are **consistent**
✅ All relationships are clearly labeled

Make any improvements before moving to Part 5.

Part 5 – Schema Normalization Practice (10 minutes)

Instructions:

1. Choose one table from your final schema.

2. Choose one normalization stage to apply: 1NF, 2NF, or 3NF.
3. Create the normalized table(s) resulting at that stage:
    - List table name(s), columns, and 3 sample rows per table.
4. Briefly explain the normalization rule(s) you applied and what changes happened.

1. Since my tables are in normalized form, I just picked the table where I applied 1NF.

TeacherID	TeacherName	SubjectCode
SENSEI 001	Gojo Satoru	CS 002
SENSEI 001	Gojo Satoru	MATH 404
SENSEI 002	Hatake Kakashi	PE 900
SENSEI 003	Rayleigh	THESIS 100

Why did I choose 1NF:

I choose 1NF in **Teachers Table** because a teacher can have multiple subjects (SubjectCode) but the **TeacherID** is still the same. I did this to make an atomic value which means every column has only a one value.

Part 6 – Relationship Analysis and Refinement (20 minutes)

Instructions:

1. Using your final schema, list **all relationships** between tables. For each, specify:
    - The two tables involved
    - The type of relationship (1:1, 1:M, M:N)
2. Identify if any **M:N (many-to-many) relationships** exist.
3. For every M:N relationship, create a **junction/bridge table** to resolve it. Provide:
    - Table name
    - Attributes (columns) with keys
    - Sample data (3 rows minimum)

Relationships:

Table A	Table B	Relationship Type
Students	Subjects	1 : M
Subjects	Teachers	M : N

Junction/Bridge Table

Enrollment Table:

EnrollmentID	Student_ID	StudentName	SubjectCode
E001	SSJ1	Naruto	THESIS 100
E002	SSJ1	Naruto	CS 002
E003	SSJ2	Sasuke	MATH 404
E004	SSJ3	Sakura	PE 900

Example template to fill in:

Table A	Table B	Relationship Type

Cheat Sheet – Quick Reference

Entity (Table) – A collection of related data about a specific object (e.g., Student, Teacher, Subject).

Attribute (Column) – A specific detail about the entity (e.g., StudentName, SubjectCode).

Primary Key (PK) – A unique identifier for each record in a table (e.g., Student_ID).

Foreign Key (FK) – An attribute that links one table to another table’ s primary key.

Relationship Types:

- **1:1** – One record in Table A matches only one record in Table B.
- **1:M** – One record in Table A can match many records in Table B.
- **M:N** – Many records in Table A can match many in Table B (requires a bridge table).

Normalization:

- **UNF** – Data may contain repeating groups.
- **1NF** – Remove repeating groups, ensure atomic values.

- **2NF** – Ensure every non-key attribute depends on the entire primary key.
- **3NF** – Remove transitive dependencies (non-key attributes depending on other non-key attributes).

Cheat Sheet: Common Database Data Types

Data Type	Description	Example Values	Use Case / Notes
INT / INTEGER	Whole numbers (no decimals)	1, 100, -50	IDs, counts, quanti
VARCHAR(n)	Variable-length text, up to n chars	"Anna", "Math101", "Mr. Cruz"	Names, codes, sho descriptions
CHAR(n)	Fixed-length text, exactly n chars	"T01 ", "S001 "	Codes or fixed-leng
TEXT	Long text data	Paragraphs, comments	Descriptions, notes
DATE	Date values	8/12/2025	Birthdates, enrollm
DATETIME	Date and time	8/12/2025 14:30	Timestamps, logs
FLOAT / DOUBLE	Decimal numbers, approximate precision	3.14, 0.001, -45.67	Prices, measureme
BOOLEAN / BOOL	True or False values	TRUE, FALSE	Status flags, yes/nc

---

Tips for Choosing Data Types:

- Use **INT** for counting and identifiers when decimals aren’ t needed.
- Use **VARCHAR** for names or codes with variable length. Keep size reasonable to save space (e.g., VARCHAR(50)).
- Use **CHAR** for fixed-length codes if all values have the same size.
- Use **TEXT** for long, variable-length content like descriptions or comments.
- Use **DATE** and **DATETIME** for time-related information to enable date functions.
- Use **BOOLEAN** for simple yes/no or true/false flags.
- Always pick the most **appropriate smallest size** to optimize storage.
